

CLAIMS:

1 1. A host-fabric adapter, comprising:
2 one or more Micro-Engines arranged to establish connections and support data transfer
3 operations, via a switched fabric, in response to work requests from a host system for data
4 transfer operations; and
5 a remote key manager arranged to manage remote keys and check the validity of the
6 remote keys which correspond to outstanding data transfer operations, via said switched fabric.

1 2. The host-fabric adapter as claimed in claim 1, further comprising a transport
2 engine which contains a plurality of work queue pairs (WQPs) in which work requests in a form
3 of descriptors are posted to describe data transfer operations and locations of data to be moved
4 for processing and/or transportation via said switched fabric.

1 3. The host-fabric adapter as claimed in claim 2, wherein said work queue pairs
2 (WQPs) each comprises:
3 a Send Queue (SQ) utilized as an "initiator" which requests normal message sends to
4 remote Virtual Interfaces (VIs) of a remote system, remote direct memory access (RDMA) reads
5 which request messages to be read from specific memory locations of said remote system, via
6 said switched fabric, and remote direct memory access (RDMA) writes which request messages
7 to be written onto specific memory locations of said remote system, via said switched fabric; and

1 a Receive Queue utilized as a "responder" which receives requests for messages from
2 normal sends, RDMA reads and RDMA writes from said remote system, via said switched
3 fabric.

1 4. The host-fabric adapter as claimed in claim 3, wherein said Micro-Engines and
2 said remote key manager are configured in accordance with the "*InfiniBandTM Specification*",
3 and implemented as part of an Application Specific Integrated Circuit (ASIC).

1 5. The host-fabric adapter as claimed in claim 3, wherein said Micro-Engines
2 comprise:

3 a Send Queue (SQ) Micro-Engine arranged to control operations of the Send Queue (SQ)
4 of the work queue pair (WQP); and

5 a Receive Queue (RQ) Micro-Engine arranged to control operations of the Receive
6 Queue (RQ) of the work queue pair (WQP).

1 6. The host-fabric adapter as claimed in claim 5, wherein said remote key manager
2 contains a remote key memory for storing stores remote keys utilized for comparison with
3 remote keys included in work requests for key validation.

1 7. The host-fabric adapter as claimed in claim 6, wherein said remote keys are used
2 to identify an appropriate page for virtual to physical address translation for outstanding data
3 transfer operations including RDMA read/write operations.

1 8. The host-fabric adapter as claimed in claim 7, wherein said work requests are
2 posted in a form of descriptors for normal data transfer operations, such as normal sends, RDMA
3 writes and RDMA reads, or in a form of Bind descriptors for non data operations, such as
4 moving or invalidating "Memory Windows" within particular memory regions of a host memory
5 of said host system on the Send Queue (SQ) to restrict RDMA data transfer operations.

1 9. The host-fabric adapter as claimed in claim 6, wherein said remote key Memory
2 is a random-access-memory (RAM) having a single read port and a single write port.

1 10. The host-fabric adapter as claimed in claim 8, wherein said remote key manager
2 is activated to check the validity of remote keys by:

3 determining if a descriptor posted at the Send Queue (SQ) is a Bind descriptor for a non
4 data operation;

5 if the descriptor posted is not a Bind Descriptor, processing for a normal data transfer
6 operation, such as a normal send, a RDMA write and a RDMA read;

7 if the descriptor posted is a Bind Descriptor, determining if the "Memory Window" is
8 invalidated;

1 if the "Memory Window" is not invalidated, noting that the "Memory Window" is moved
2 and performing writes to the host memory of said host system to move the "Memory Window";
3 if the "Memory Window" is invalidated, performing writes to the host memory of said
4 host system to destroy the "Memory Window";
5 searching through all outstanding RDMA operations which use the "Memory Window"
6 to identify a remote key which used the "Memory Window";
7 if the remote key which used the "Memory Window" is identified, invalidating the
8 remote key until all remote keys are invalidated;
9 if no remote key is identified, completing invalidation of all the remote keys and
10 returning to being idle; and
11 if all remote keys are not invalidated, returning to search through all outstanding RDMA
12 operations which uses the "Memory Window" until all remote keys are marked "invalidated" so
13 that no new remote key or new work queue pair (WQP) can come in and use that "Memory
14 Window".

1 11. The host-fabric adapter as claimed in claim 10, wherein said remote key manager
2 is activated upon a request for key invalidation from said SQ Micro-Engine to invalidate remote
3 keys by:
4 initiating a Protection Domain (PD) compare between the Memory PD and SQ supplied
5 PD of a Virtual Interface (VI);

1 if the Memory PD does not match the SQ supplied PD, skipping processing the remote
2 keys for that VI, and determining whether all VIs are done; and
3 if the Memory PD matches the SQ supplied PD, reading and initiating remote key
4 compares sequentially for all remote keys associated with that VI for key invalidation.

1 12. The host-fabric adapter as claimed in claim 11, wherein said remote key manager
2 initiates remote key compares for all remote keys associated with that VI by:

3 reading and comparing a first remote key from the remote key Memory with the SQ
4 supplied key;

5 if the first remote key does not match the SQ supplied key, moving to check a next
6 remote key;

7 if the first remote key matches the SQ supplied key, clearing a Valid bit corresponding to
8 the first remote key in the remote key Memory;

9 reading and comparing a second remote key from the remote key Memory with the SQ
10 supplied key;

11 if the second remote key does not match the SQ supplied key, moving to check the next
12 remote key;

13 if the second remote key matches the SQ supplied key, clearing the Valid bit
14 corresponding to the second remote key in the remote key Memory;

15 reading and comparing a third remote key from the remote key Memory with the SQ
16 supplied key;

1 if the third memory remote key does not match the SQ supplied key, moving to check a
2 next remote key;

3 if the third remote key matches the SQ supplied key, clearing the Valid bit corresponding
4 to third remote key in the remote key Memory;

5 reading and comparing a last remote key from the remote key Memory with the SQ
6 supplied key;

7 if the last memory remote key does not match the SQ supplied key, moving to determine
8 whether all VIs are done; and

9 if the last remote key matches the SQ supplied key, clearing the Valid bit corresponding
10 to the last remote key in the remote key Memory and determining whether all VIs are done.

1 13. A host-fabric adapter installed at a host system for connecting to a switched fabric
2 of a data network, comprising:

3 at least one Micro-Engine (ME) arranged to establish connections and support data
4 transfers via said switched fabric;

5 a serial interface arranged to receive and transmit data packets from said switched fabric
6 for data transfer operations;

7 a host interface arranged to receive and transmit work requests, in the form of
8 descriptors, from said host system for data transfer operations;

9 a context memory arranged to store context information needed for said Micro-Engine
10 (ME) to process work requests for data transfer operations;

1 a doorbell manager arranged to update the context information needed for said Micro-
2 Engine (ME) to process work requests for data transfer operations; and
3 a remote key manager arranged to manage remote keys and check the validity of the
4 remote keys which correspond to outstanding data transfer operations.

1 14. The host-fabric adapter as claimed in claim 13, further comprising a transport
2 engine which contains a plurality of work queue pairs (WQPs) in which work requests in a form
3 of descriptors are posted to describe data transfer operations and locations of data to be moved
4 for processing and/or transportation via said switched fabric.

1 15. The host-fabric adapter as claimed in claim 14, wherein said work queue pairs
2 (WQPs) each comprises:

3 a Send Queue (SQ) utilized as an "initiator" which requests normal message sends to
4 remote Virtual Interfaces (VIs) of a remote system, remote direct memory access (RDMA) reads
5 which request messages to be read from specific memory locations of said remote system, via
6 said switched fabric, and remote direct memory access (RDMA) writes which request messages
7 to be written onto specific memory locations of said remote system, via said switched fabric; and

8 a Receive Queue utilized as a "responder" which receives requests for messages from
9 normal sends, RDMA reads and RDMA writes from said remote system, via said switched
10 fabric.

1 16. The host-fabric adapter as claimed in claim 13, wherein said Micro-Engine, said
2 host interface, said serial interface, said context memory, said doorbell manager, and said remote
3 key manager are configured in accordance with the "*InfiniBandTM Specification*", and
4 implemented as part of an Application Specific Integrated Circuit (ASIC).

1 17. The host-fabric adapter as claimed in claim 15, wherein said Micro-Engine
2 includes:

3 a Send Queue (SQ) Micro-Engine arranged to control operations of the Send Queue (SQ)
4 of the work queue pair (WQP); and

5 a Receive Queue (RQ) Micro-Engine arranged to control operations of the Receive
6 Queue (RQ) of the work queue pair (WQP).

1 18. The host-fabric adapter as claimed in claim 17, wherein said remote key manager
2 contains a remote key memory for storing stores remote keys utilized for comparison with
3 remote keys included in work requests for key validation.

1 19. The host-fabric adapter as claimed in claim 18, wherein said remote keys are used
2 to identify an appropriate page for virtual to physical address translation for outstanding data
3 transfer operations including RDMA read/write operations.

1 20. The host-fabric adapter as claimed in claim 19, wherein said work requests are
2 posted in a form of descriptors for normal data transfer operations, such as normal sends, RDMA
3 writes and RDMA reads, or in a form of Bind descriptors for non data operations, such as
4 moving or invalidating "Memory Windows" within particular memory regions of a host memory
5 of said host system on the Send Queue (SQ) to restrict RDMA data transfer operations.

1 21. The host-fabric adapter as claimed in claim 18, wherein said remote key Memory
2 is a 1280x28 random-access-memory (RAM) having a single read port and a single write port.

1 22. The host-fabric adapter as claimed in claim 20, wherein said remote key manager
2 is activated to check the validity of remote keys by:

3 determining if a descriptor posted at the Send Queue (SQ) is a Bind descriptor for a non
4 data operation;

5 if the descriptor posted is not a Bind Descriptor, processing for a normal data transfer
6 operation, such as a normal send, a RDMA write and a RDMA read;

7 if the descriptor posted is a Bind Descriptor, determining if the "Memory Window" is
8 invalidated;

9 if the "Memory Window" is not invalidated, noting that the "Memory Window" is moved
10 and performing writes to the host memory of said host system to move the "Memory Window";

11 if the "Memory Window" is invalidated, performing writes to the host memory of said
12 host system to destroy the "Memory Window";

1 searching through all outstanding RDMA operations which use the "Memory Window"
2 to identify a remote key which used the "Memory Window";

3 if the remote key which used the "Memory Window" is identified, invalidating the
4 remote key until all remote keys are invalidated;

5 if no remote key is identified, completing invalidation of all the remote keys and
6 returning to being idle; and

7 if all remote keys are not invalidated, returning to search through all outstanding RDMA
8 operations which uses the "Memory Window" until all remote keys are marked "invalidated" so
9 that no new remote key or new work queue pair (WQP) can come in and use that "Memory
10 Window".

1 23. The host-fabric adapter as claimed in claim 22, wherein said remote key manager
2 is activated upon a request for key invalidation from said SQ Micro-Engine to invalidate remote
3 keys by:

4 initiating a Protection Domain (PD) compare between the Memory PD and SQ supplied
5 PD of a Virtual Interface (VI);

6 if the Memory PD does not match the SQ supplied PD, skipping processing the remote
7 keys for that VI, and determining whether all VIs are done; and

8 if the Memory PD matches the SQ supplied PD, reading and initiating remote key
9 compares sequentially for all remote keys associated with that VI for key invalidation.

1 24. The host-fabric adapter as claimed in claim 23, wherein said remote key manager
2 initiates remote key compares for all remote keys associated with that VI by:

3 reading and comparing a first remote key from the remote key Memory with the SQ
4 supplied key;

5 if the first remote key does not match the SQ supplied key, moving to check a next
6 remote key;

7 if the first remote key matches the SQ supplied key, clearing a Valid bit corresponding to
8 the first remote key in the remote key Memory;

9 reading and comparing a second remote key from the remote key Memory with the SQ
10 supplied key;

11 if the second remote key does not match the SQ supplied key, moving to check the next
12 remote key;

13 if the second remote key matches the SQ supplied key, clearing the Valid bit
14 corresponding to the second remote key in the remote key Memory;

15 reading and comparing a third remote key from the remote key Memory with the SQ
16 supplied key;

17 if the third memory remote key does not match the SQ supplied key, moving to check a
18 next remote key;

19 if the third remote key matches the SQ supplied key, clearing the Valid bit corresponding
20 to third remote key in the remote key Memory;

1 reading and comparing a last remote key from the remote key Memory with the SQ
2 supplied key;
3 if the last memory remote key does not match the SQ supplied key, moving to determine
4 whether all VIs are done; and
5 if the last remote key matches the SQ supplied key, clearing the Valid bit corresponding
6 to the last remote key in the remote key Memory and determining whether all VIs are done.

1 25. A method of checking the validity of remote keys which correspond to
2 outstanding remote direct memory access (RDMA) operations in a host-fabric adapter installed
3 at a host system, comprising:
4 determining if any Virtual Interface (VI) to be processed at a Send Queue (SQ) is a Bind
5 descriptor for a non data operation;
6 if the descriptor posted is not a Bind Descriptor, processing for a normal data transfer
7 operation;
8 if the descriptor posted is a Bind Descriptor, determining if the "Memory Window" is
9 invalidated;
10 if the "Memory Window" is not invalidated, noting that the "Memory Window" is moved
11 and performing writes to a host memory of said host system to move the "Memory Window";
12 if the "Memory Window" is invalidated, performing writes to the host memory of said
13 host system to destroy the "Memory Window";

1 searching through all outstanding RDMA operations which use the "Memory Window"
2 to identify a remote key which used the "Memory Window";

3 if the remote key which used the "Memory Window" is identified, invalidating the
4 remote key until all remote keys are invalidated;

5 if no remote key is identified, completing invalidation of all the remote keys and
6 returning to being idle; and

7 if all remote keys are not invalidated, returning to search through all outstanding RDMA
8 operations which uses the "Memory Window" until all remote keys are marked "invalidated" so
9 that no new remote key or new work queue pair (WQP) can come in and use that "Memory
10 Window".

1 26. The method as claimed in claim 25, wherein said remote keys are invalidated by:
2 initiating a Protection Domain (PD) compare between a Memory PD and SQ supplied PD
3 of a Virtual Interface (VI);

4 if the Memory PD does not match the SQ supplied PD, skipping processing the remote
5 keys for that VI, and determining whether all VIs are done; and

6 if the Memory PD matches the SQ supplied PD, reading and initiating remote key
7 compares sequentially for all remote keys associated with that VI for key invalidation.

1 27. The method as claimed in claim 26, wherein said remote key compares for all
2 remote keys associated with that VI are performed by:

1 reading and comparing a first remote key from a remote key Memory with the SQ
2 supplied key;

3 if the first remote key does not match the SQ supplied key, moving to check a next
4 remote key;

5 if the first remote key matches the SQ supplied key, clearing a Valid bit corresponding to
6 the first remote key in the remote key Memory;

7 reading and comparing a second remote key from the remote key Memory with the SQ
8 supplied key;

9 if the second remote key does not match the SQ supplied key, moving to check the next
10 remote key;

11 if the second remote key matches the SQ supplied key, clearing the Valid bit
12 corresponding to the second remote key in the remote key Memory;

13 reading and comparing a third remote key from the remote key Memory with the SQ
14 supplied key;

15 if the third memory remote key does not match the SQ supplied key, moving to check a
16 next remote key;

17 if the third remote key matches the SQ supplied key, clearing the Valid bit corresponding
18 to third remote key in the remote key Memory;

19 reading and comparing a last remote key from the remote key Memory with the SQ
20 supplied key;

1 if the last memory remote key does not match the SQ supplied key, moving to determine
2 whether all VIs are done; and
3 if the last remote key matches the SQ supplied key, clearing the Valid bit corresponding
4 to the last remote key in the remote key Memory and determining whether all VIs are done.